

Flexbox

CSS Flexible Box Layout module, ili skraćeno ***Flexible Box***, ***Flexbox*** ili najjednostavnije samo ***Flex*** je sistem osmišljen kako bi kreirao jednostavniji i efikasniji način raspoređivanja, poravnanja i razmeštanja elemenata u okviru određenog okvira, čak i ukoliko dimenzije nisu definisane i/ili trebaju biti dinamički određene spram određenih parametara, bilo zadatim ili ne.

drugim rečima..

Primarni cilj i ideja flex-a je da se omogući okviru da menja dimenzije elemenata unutar sebe (njegove dece) kako bi na najbolji način popunili dostupan prostor.

Zvanična W3C specifikacija samog Flex-a kaže:

“In the flex layout model, the children of a flex container can be laid out in any direction, and can “flex” their sizes, either growing to fill unused space or shrinking to avoid overflowing the parent. Both horizontal and vertical alignment of the children can be easily manipulated. Nesting of these boxes (horizontal inside vertical, or vertical inside horizontal) can be used to build layouts in two dimensions.”

<https://www.w3.org/TR/css-flexbox/>

Flex container (flex okvir)

Objekat koji koristimo kao osnovu za raspoređivanje elemenata koristeći flex (flexbox) se naziva flex container, odnosno u prevodu, flex okvir. Da bi kreirali flex okvir koristimo već dobro poznato svojstvo ***display***, ali sada sa vrednošću ***flex***, odnosno ***inline-flex***.

```
.object {  
    display: flex;  
}
```

Na taj način, sam objekat koji je pogođen, u zavisnosti od vrednosti, ostaje block ili inline element, ali njegova direktna deca odmah i automatski postaju ***flex items***, odnosno u prevodu ***flex elementi***.

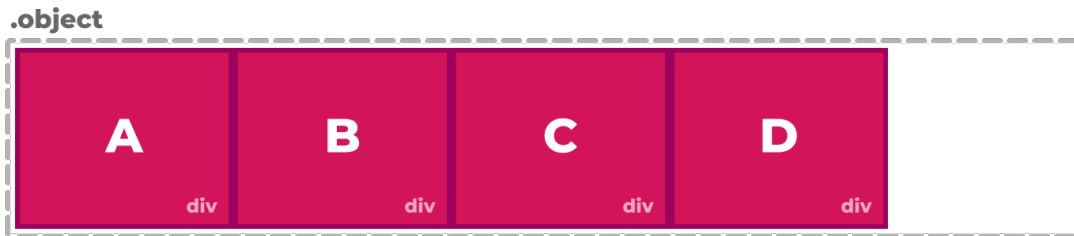
Drugim rečima, menjajući prirodu roditelja i dodeljujući flex osobine roditelju, deca automatski, bez ikakvog dodeljivanja svojstva ili vrednosti postaju flex elementi.

Kao i kod većine drugih svojstava i vrednosti, postavljene su neke default, odnosno osnovne osobine i vrednosi.

- Svi elementi će biti prikazani u jednom redu (horizontalno) i nijedan neće preći u novi red. Ukoliko se ne mogu smestiti, tj ukoliko nema dovoljno prostora horizontalno, izaćiće van okvira (overflow).
- Elementi kreću od početka osnovne ose (main axis) i ređaju se redom prema njenom kraju. Kod LTR jezika početak je na levoj strani, kraj na desnoj.
- Elementi se neće širiti (stretch) u okviru osnovne ose (horizontalno po default podešavanjima) ali se mogu smanjiti po istoj (shrink)
- Elementima će se povećati visina kako bi ispunili kompletnu dimenziju poprečne ose (cross axis). U osnovi, po default vrednostima, to je visina i vertikalna osa.

```
<div class="object">  
  <div>A</div>  
  <div>B</div>  
  <div>C</div>  
  <div>D</div>  
</div>
```

```
.object {  
  display: flex;  
}
```



Flexbox ose

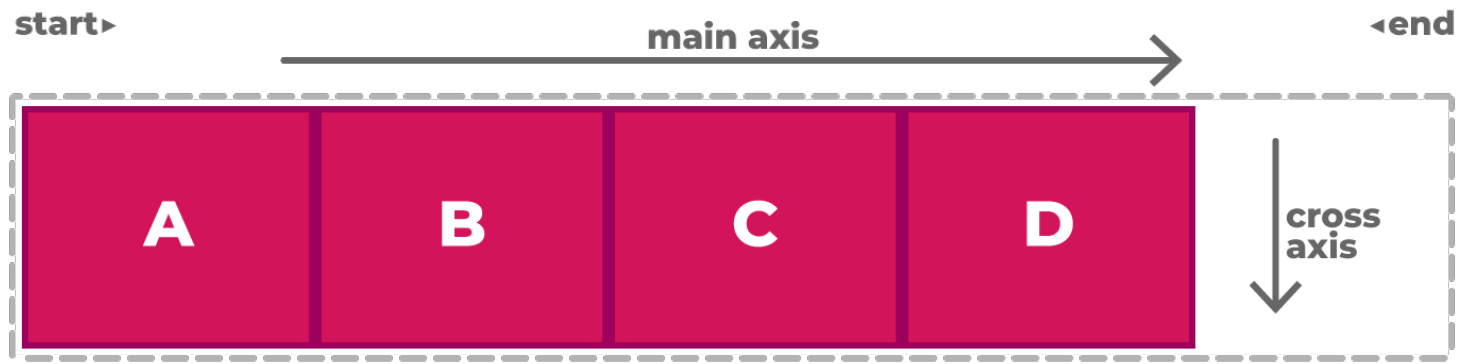
- Kada radimo sa Flex elementima, uvek moramo imati na umu dve ose: **glavna osa (main axis)** i **poprečna osa (cross axis)**.
- Glavna osa se po default vrednostima prostire horizontalno sa početkom na levoj strani, a kraj na desnoj. Poprečna osa se, kao što i ime govori, nalazi poprečno u odnosu na glavnu osu (ugao od 90°) i njen početak je na vrhu, a kraj na dnu.

flex-direction

Prostiranje glavne ose (main axis) definišemo svojstvom ***flex-direction*** koje može imati jednu od četiri vrednosti:

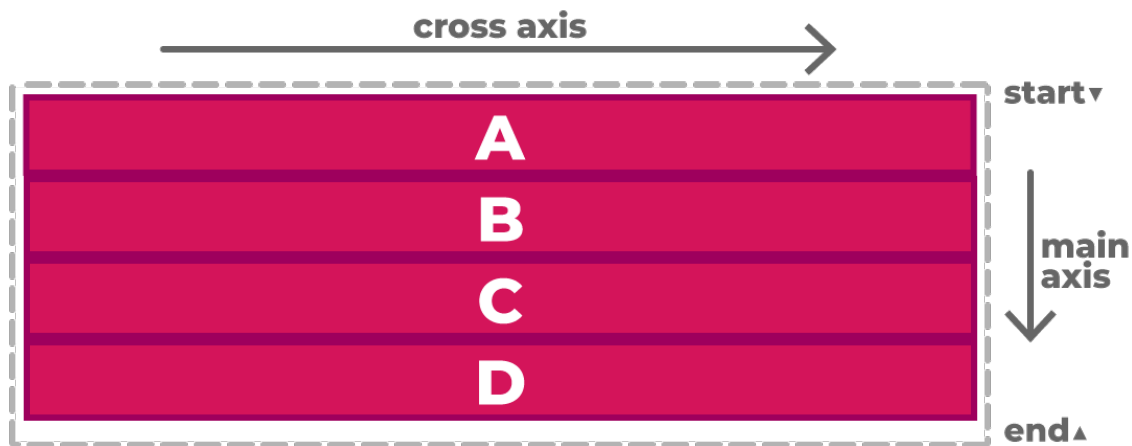
- **row** – Default vrednost. Osnovna osa se prostire sa leve na desnu stranu
- **row-reverse** – Osnovna osa se prostire horizontalno ali sa desne na levu stranu
- **column** – Osnovna osa se prostire vertikalno, sa početkom na vrhu
- **column-reverse** - Osnovna osa se prostire vertikalno, sa početkom na dnu

flex-direction



flex-direction

flex-direction: column



Flex u više redova (*flex-wrap*)

- Po osnovnim default podešavanjima, flex container će pokušati da smesti sve svoje elemente (direktnu decu) u jednom vidljivom redu
- Ukoliko je okvir dovoljno širok (ili visok u slučaju promene orijentacije osa) za sve elemente, neće biti problema.
- Ipak, ukoliko to nije slučaj, možemo dobiti ili veoma uske elemente ili će doći do overflow ponašanja. Iako je flex jednodimenzionalna struktura primarno, možemo naterati flex elemente da pređu u nove redove (eng. *Wrap*). U tom slučaju svaki red (paralelan sa osnovnom osom) će biti tretiran kao **zasebni flex okvir**. Ukoliko dolazi do raspoređivanja praznog prostora i/ili širina, to će se vršiti samo u okviru tog reda, bez uticaja na, i sa, drugih redova.

Eventualno prostiranje u okviru više redova, podešavamo svojstvom ***flex-wrap*** na okviru. Moguće vrednosti su:

- **nowrap** – Default vrednost. Svi flex elementi su u jednom redu.
- **wrap** - Flex elementi će prelaziti u nove redove po potrebi, od početka osnovne ose ka kraju.
- **wrap-reverse** - Flex elementi će prelaziti u nove redove po potrebi, od kraja osnovne ose ka početku.

- Iako možemo flex-direction i flex-wrap pisati zasebno, možemo ih i skratiti i pisati zajedno pod **flex-flow** svojstvom.
- U okviru njega, prva komponenta vrednosti će biti flex-direction, dok će druga biti flex-wrap.

Umesto, na primer:

```
.object {  
    display: flex;  
    flex-direction: row;  
    flex-wrap: wrap;  
}
```

Možemo:

```
.object {  
    display: flex;  
    flex-flow: row wrap;  
}
```

Redosled elemenata (order)

- Po osnovnim default podešavanjima, flex elementi će biti prikazani u redosledu kako su postavljeni u HTML strukturi, prateći pravac i smer osnovne ose flex okvira.
- Drugim rečima, kako bi i očekivali.

Možemo na nivou samih elemenata određivati njihov redosled, koristeći svojstvo **order**.

```
<div class="object">  
<div class="item1"> 1 </div>  
<div class="item2"> 2 </div>  
<div class="item3"> 3 </div>  
<div class="item4"> 4 </div>  
<div class="item5"> 5 </div>  
</div>
```

```
.object {display: flex;}  
.item1 { order: 2; }  
.item2 { order: 1; }  
.item3 { order: 4; }  
.item4 { order: 5; }  
.item5 { order: 3; }
```

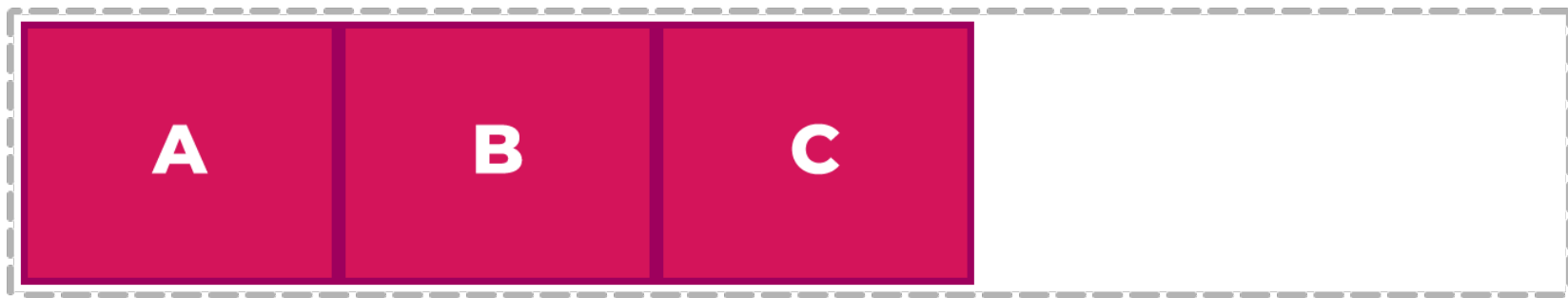


Ponašanje flex elemenata

- Po osnovnim default podešavanjima, ukoliko im nisu definisane širine, flex elementi će imati širinu adekvatnu sadržaju.
- Imajte na umu da, iako su oni možda inicijalno bili *block* elementi (poput div-ova), smeštanje u flex okviru i pretvaranje u flex elemente, menja njihovo ponašanje.

Ponašanje flex elemenata

flex container width $>$ A + B + C width



flex-basis

Ovo svojstvo je veoma slično klasičnoj širini definisanoj preko width svojstva. Ipak, dok je širina uvek na istoj (horizontalnoj) osi, flex-basis prati osnovnu osu flexbox-a i iako je po default podešavanjima horizontalna, to ne mora uvek biti slučaj.

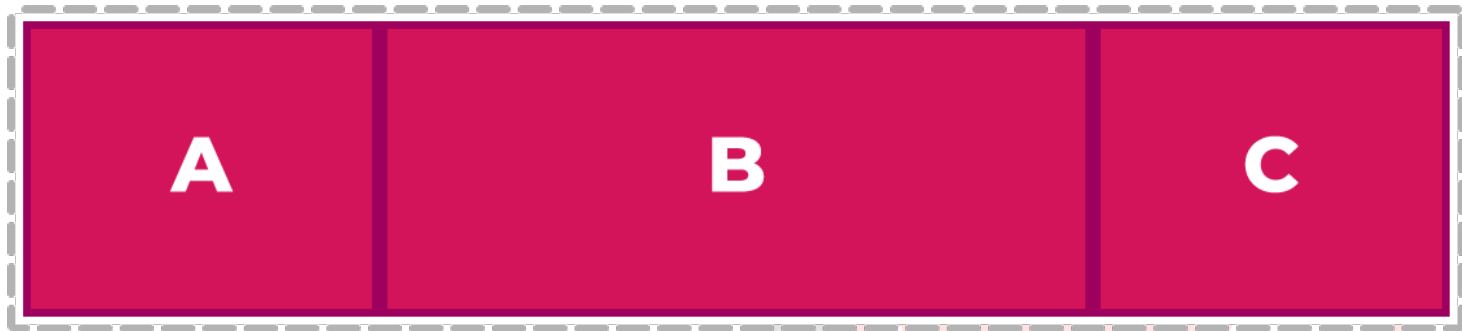
flex-grow

Ovo svojstvo definiše da li flex elementima dozvoljavamo da povećavaju svoju dimenziju prateći osnovnu flex osu (najčešće širinu) kako bi popunili dostupan prostor. Default vrednost je 0 i elementi neće odstupiti od svoje flex-basis vrednosti.

flex-grow

- Moguće vrednosti ovog svojstva su celi brojevi. Ukoliko postavimo vrednost 1 određenom elementu, on će dobiti mogućnost da se po potrebi proširi. Ukoliko svim flex elementima postavimo vrednost ovog svojstva na 1, oni će odstupati od svoje osnovne flex-basis vrednosti i podjednako rasporediti preostalo dostupan prostor na osnovnoj osi.
- Flex-grow možemo koristiti i da proporcionalno rasporedimo prostor i elemente. Na primer, ukoliko postavimo tri flex elementa u flex okviru i za dva zadamo flex-grow sa vrednošću 1, a jednom sa vrednošću 2, on će se potruditi da bude duplo širi od njih.

```
.object {  
    display: flex;  
    flex-flow: row nowrap;  
}  
.itemA { flex-grow: 1; }  
.itemB { flex-grow: 2; }  
.itemC { flex-grow: 1; }
```



flex-shrink

Dok flex-grow svojstvo utiče na potencijalno povećanje, flex-shrink radi upravo suprotno na veoma sličan način, odnosno utiče na to kako se elementu oduzima prostor. Default vrednost je 1. Upotrebom ovog svojstva, element može biti manji nego zadata flex-basis vrednost. Takođe može biti definisan u proporciji.

Skraćeno pisanje grow, shrink i basis vrednosti

Umesto zasebnog pisanja flex-grow, flex-shrink i flex-basis vrednosti, možemo ih skratiti i pisati u jednom svojstvu zajedno (takozvano eng. *Shorthand*) koristeći jednostavno svojstvo ***flex***.

Ovo je i **preporučeni način**.

Na primer, umesto:

```
.item {  
    flex-grow: 0;  
    flex-shrink: 1;  
    flex-basis: 180px;  
}
```

možemo pisati:

```
.item {  
    flex: 0 1 180px;  
}
```

flex-start



flex-end



center



space-between



space-around



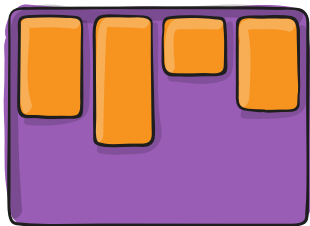
space-evenly



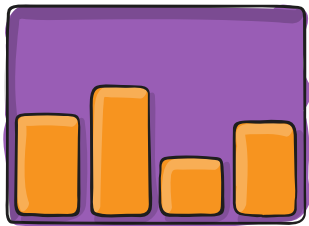
Justify-content

Definiše poravnanje po glavnoj osi.

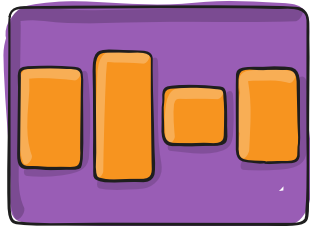
flex-start



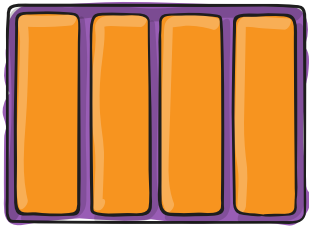
flex-end



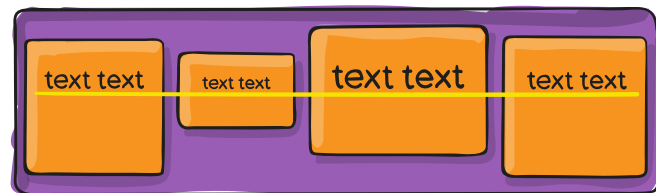
center



stretch



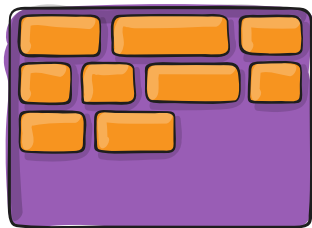
baseline



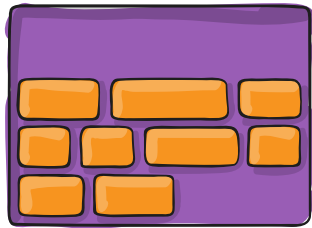
align-items

Definiše poravnanje po sporednoj, poprečnoj osi.

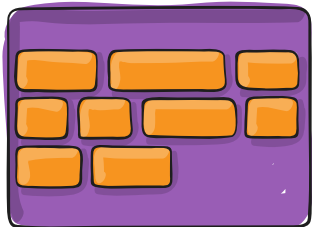
flex-start



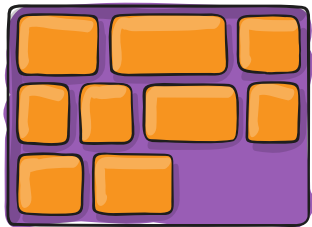
flex-end



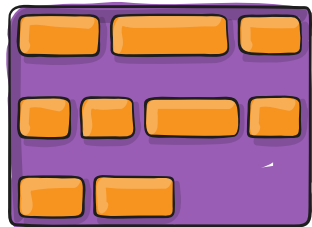
center



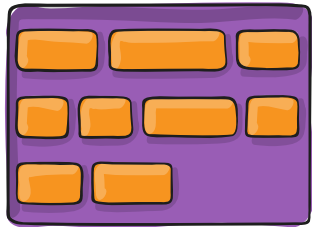
stretch



space-between



space-around



align-content

Ukoliko postoji više redova flex sadržaja definiše pozicije i prostor između njih.

align-self (*flex-item*)

Funkcioniše poput *align-items*, ali se za razliku od njega (koji se potavlja na okviru) **align self postavlja na pojedinačnom flex elementu.**

Iste su vrednosti.

